McGinn & Gibb, P.C.
A Professional Limited Liability Company
Patents, Trademarks, Copyrights, and Intellectual Property Law
1701 Clarendon Boulevard, Suite 100
Arlington, Virginia 22209
Telephone (703) 294-6699
Facsimile (703) 294-6696

APPLICATION FOR UNITED STATES LETTERS PATENT

APPLICANTS:	German Goldszmidt, Jean A. Lorrain, Kiyoshi Maruyama, and Dinesh Chandra Verma
FOR:	METHOD AND APPARATUS FOR DYNAMICALLY ADJUSTING RESOURCES ASSIGNED TO PLURALITY OF CUSTOMERS, FOR MEETING SERVICE LEVEL AGREEMENTS (SLAs) WITH MINIMAL RESOURCES, AND ALLOWING COMMON POOLS OF RESOURCES TO BE USED ACROSS PLURAL CUSTOMERS ON A DEMAND BASIS
DOCKET NO.:	YOR999-479

METHOD AND APPARATUS FOR DYNAMICALLY ADJUSTING RESOURCES ASSIGNED TO PLURALITY OF CUSTOMERS, FOR MEETING SERVICE LEVEL AGREEMENTS (SLAs) WITH MINIMAL RESOURCES, AND ALLOWING COMMON POOLS OF RESOURCES TO BE USED ACROSS PLURAL CUSTOMERS ON A DEMAND BASIS

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to a world-wide network, and more particularly to sites of a plurality of Internet World Wide Web (WWW) sites of various owners hosted by a service provider using a group of servers and meeting with agreed-upon service levels.

Description of the Related Art

The Internet is the world's largest network, and it has become essential to businesses as well as to consumers. Many businesses have started out-sourcing their e-business and e-commerce Web sites to service providers, instead of operating their Web sites on their own server(s) and managing them by themselves. Such a service provider must install a collection of servers in a farm called a "Web Server Farm (WSF)", or a "Universal Server Farm (USF)" which can be used by many different businesses to run their e-commerce and e-business applications.

These business customers (e.g., the service provider's "customers") have different "server resource" requirements for their Web sites and applications.

When businesses (hereafter referred to as "customers" or "customers of a server farm") out-source their e-commerce and/or e-business to a service provider, they must obtain some guarantee on the services they are getting (and will continue to obtain) from the service provider for their sites. Once the service provider has made a commitment to a customer to provide a certain "level" of service (e.g., referred to as a "Service Level Agreement (SLA)"), the provider must guarantee that level of service to that customer.

Figure 1 illustrates an abstracted view of a conventional server farm. A server farm 103 includes multiple servers which host customer applications, and is connected to Internet 101 via communications link(s) 102. Each customer's server resource requirements changes since the demands to customers' applications change continuously on a dynamic basis during each day of operations.

However, a problem with the conventional system and method used thereby is that, hitherto the present invention, there has been no provision for dynamically equipping the server farm such that server(s) and their resources can be dynamically allocated. Hence, there has been no flexibility in dynamically allocating servers and their resources to customers as the customer's demands change. This results in system-wide inefficiency and general dissatisfaction by the customer.

Another problem with the conventional system is that there are no Service Level Agreements (SLAs) based on dynamic allocation and de-allocation of servers to customer's server clusters.

Yet another problem with the conventional system is that there is no provisioning of SLAs in support of both a guaranteed number of servers and optional additional servers based on

the workload changes to customers' applications. Yet another problem with the conventional system is that a "hacker" or "hackers" can generate a large amount of workload to a customer's sites or to the server farm itself to "crash" servers or server farm.

SUMMARY OF THE INVENTION

In view of the foregoing and other problems of the conventional methods and structures, an object of the present invention is to provide a method and structure in which an allocation of server resources for a plurality of customers is dynamically controlled.

Another object of the present invention is to support the (minimum, maximum) server resource-based service level agreements for a plurality of customers.

Yet another object of the present invention is to control the allocation of additional server resources to a plurality of customers using the bounds on given service level metrics.

Still another object of the present invention is to support various service level metrics.

A further object of the present invention is to support the use of different metrics for different customers.

Another object of the present invention is to use a service level metric, the amount of allocated resources, and the inbound traffic rate, for defining the state of the current service level (M,N,R) for each customer.

Another object of the present invention is to use a "target" service level metric Mt to keep the actual service level M close to the target service level.

A further object of the present invention is to compute a "target" amount of resources Nt and the inbound traffic rate Rt from a given Mt and (M,N,R).

Still another object of the present invention is to provide and use formulas for computing Nt and Rt from Mt and (M,N,R).

A still further object of the present invention is to allow the use of numerical analysis or quick simulation techniques for deriving Nt and Rt in place of using formulas invented and described in this patent application.

Yet another object of the present invention is to support resource utilization U for M, average response time T for an actual service level M, and the response time percentile T% for the actual service level M (and therefore, the support of targets Ut, Tt and Tt%).

Another object of the present invention is to provide a method (decision algorithm) for deciding whether or not to add additional server resource(s) or to reduce ("throttle down") the inbound traffic to meet the service level agreements for a plurality of customers.

In a first aspect of the present invention, a method (and system) for managing and controlling allocation and de-allocation of resources based on a guaranteed amount of resource and additional resources based on a best effort for a plurality of customers, includes dynamically allocating server resources for a plurality of customers, such that the resources received by a customer are dynamically controlled and the customer receives a minimum (e.g., a minimum that is guaranteed) amount of resources as specified under a service level agreement (SLA).

In another aspect, a program storage device is provided for storing the program of the inventive method.

With the unique and unobvious features of the present invention, a server farm is equipped with a means to dynamically allocate servers (or server resources) to customers as demands change.

It is noted that a general service level agreement (SLA) on a server resource for a customer can be denoted by (Smin#(i), Smax#(i), Mbounds(i)), where Smin#(i) denotes the guaranteed minimum amount of server resources (e.g., the number of servers), Smax(i) denotes the upper bound on the amount of server resources that a customer may want to obtain when free resources are available, and Mbounds(i) gives two bounds: Mhighbound(i) and Mlowbound(i) on a service level metric M that is used in controlling the allocation of resources beyond the minimum for each i-th customer. Mhighbound(i) is used to decide when to add additional server resources and Mlowbound (i) is used to decide when to remove some server resources.

The minimum (or min) amount of server resources (e.g., number of servers) Smin#(i) is a guaranteed amount of server resources that the i-th customer will receive regardless of the server resource usage. The maximum (or max) amount of server resources Smax#(i) is the upper bound on the amount of server resources that the i-th customer may receive beyond the minimum provided that some unused server resources are available for allocation.

Therefore, the range between Smin#(i) and Smax#(i) represents server resources that are provided on an "as-available" or "best-effort" basis, and it is not necessarily guaranteed that the customer will obtain these resources at any one time, if at all. The allocation of additional resource(s) is performed so as to keep the performance metric within Mbounds(i).

Examples of Mbounds(i) include: (1) the bound on the server resource utilization that is denoted by Ubounds(i); (2) the bound on the average server response time that is denoted by Tbounds(i); and (3) the bound on the server response time percentile that is denoted by T%bounds(i).

Table 1 provides definitions and notations used throughout the present application. For example, when Mbounds(i) = Ubounds(i) = (Ulowbound(i),Uhighbound(i) = (50%, 80%), the

server farm tries to allocate additional server resources (or de-allocate some servers) to the i-th customer's server complex to keep the server resource utilization between 50% and 80%.

That is, when the server resource utilization goes above 80%, the server farm tries to keep the utilization below 80% by allocating additional server resources to the i-th customer when free resources are available. If free resources are not available, the server farm may need to limit the amount of incoming traffic to the i-th customer's server complex. Conversely, when the server resource utilization goes below 50%, the server farm tries to remove some server resources from the i-th customer in order to keep the utilization above 50%. In order to keep the observed metric M within the given Mbounds, the notion of a "target" metric Mt is introduced. Mt is a value that falls between Mlowbound and Mhighbound and the system of the present invention tries to keep the observed metric M as close as possible to the target metric Mt by adjusting server resources. In general, the unit cost of the server resources above the minimum guarantee is more than or equal to that of the server resources below the minimum.

Thus, the present invention provides a dynamic resource allocation to a plurality of customers to meet with the (min, max) server resources and performance metric based service level agreements. Unused (un-allocated) server resources are pooled and allocated and de-allocated from the pool, thus providing sharing of server resources among plurality of customer, leading to efficient use of server resources. Since incoming workload is regulated when it has exceeded server resources allocated, the system provides a "denial of services" to some workloads, thus preventing a crash of hosted customer sites and preventing a crash of the server farm itself.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other purposes, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

Figure 1 illustrates an abstracted view of a conventional server farm;

Figure 2 illustrates a general overview of the operation and structure of the present invention;

Figure 3 illustrates a concept of a Service Level Agreement (Smin#, Smax#, Mbounds);

Figure 4 illustrates a graph showing the relationship of Metric M to the number of server resources, to show a concept of the present invention;

Figure 5 illustrates an overall system 500 and environment of the present invention; and

Figure 6 illustrates a decision method 600 for server allocation.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

Referring now to the drawings, and more particularly to Figures 1-6, there is shown a preferred embodiment of the method and structure according to the present invention.

PREFERRED EMBODIMENT

Referring to Figure 2, prior to describing the details of the invention, an overview and a primary object of the present invention will be described below.

As shown in Figure 2, the invention first monitors the inbound traffic rate R(i) 206, the currently assigned amount of server resources N(i) 205, and the current service level metric M(i) 204 for all customers 201 and 202.

Then, the inventive system performs the following actions only when M(i) falls outside of Mbounds(i), namely either M(i) is above Mhighbound(i) or M(i) is below Mlowbound(i), to avoid "allocation/de-allocation swings".

The "target" amount of server resources Nt(i), without changing the inbound traffic R(i), is computed. Further, the "target" inbound traffic rate Rt(i), without changing the allocated resource N(i), is computed in order to bring the service level metric M(i) close to the "targeted" service level metric Mt(i) from monitored R(i), N(i) and M(i) for all i. The target service level metric Mt(i) is the service level metric at or near which one wants to keep M(i) so that M(i) falls within Mbounds(i) = (Mlowbound(i),Mhighbound(i)).

Once Nt(i) and Rt(i) are computed, then it is decided how to move current M(i) to the target Mt(i), by either changing N(i) to Nt(i) (e.g., this involves either allocating server resources from free resource pool 203 to a customer's server set 201 or 202, or taking some server resources away from customer 201 or 202 and return to the pool 203) or by bounding the inbound traffic rate R(i) to Rt(i) (e.g., this is performed when either the maximum amount of resources has been already allocated or no free resource is available so that the only way to bring M(i) to Mt(i) is to reduce the amount of inbound traffic).

Once the decision has been made, it will then send a request to an appropriate systems resource manager (e.g., a "resource allocation manager" or an "inbound traffic controller").

Figure 3 illustrates the concept of the service level agreement (SLA) that the present invention supports for a plurality of customers. The service level agreement for each customer has the form of (Smin#, Smax#, Mbounds), where Smin# is the guaranteed amount of server resources (e.g., the number of servers), Smax# is the upper bound on the total amount of server resources that a customer may obtain when free resources are available, and Mbounds is a pair of bounds on the service level metric that are used in determining when to add additional resources or to remove some resources away. For ease of illustration, in Figure 3, the server resource is assumed to have (reside in) a single dimension. However, this could be a vector.

Figure 3 shows six operation spaces: A 301, B 302, C 303, D 304, E 305 and F 306. Because of the bounds Smin# 314, and Smax# 313, the feasible operation spaces are B 302 and E 305.

It is noted that the operation space D 304 could be made available especially when a server farm operator could "borrow" some servers from some customers when the customers are not fully utilizing their resources.

The operation space B 302 is a "non-desirable" space since the service level metric M is exceeding the bound Mhighbound 311. The operation space E 305 is the space in which the operational state should be kept. Furthermore, the upper portion of the space E 305 that is bounded by Mlowbound 312 and Mhighbound 311 is the operation space allowed by the exemplary service level agreement (SLA) that the present invention supports. It is noted that the metric M may be utilization, average response time, percentile response time, etc. Mbounds 307 may be Ubounds, Tbounds, T%bounds, etc. as suitably determined by the designer given constraints and requirements imposed thereon.

Figure 4 illustrates a primary concept of the present invention. Here, the operation space 305 is divided into two regions. A first region is called a "green belt" 405 (e.g., the region

bounded by Mlowbound 312 and Mhighbound 311), and a second region is the remaining space of the space 305.

In the present invention, the operation state which falls into the green belt 405 is deemed to be acceptable while the operation which falls outside of the green belt (e.g., below the green belt), is not acceptable since too many unnecessary resources are allocated, thereby incurring extra (wasteful) costs to a customer.

Figure 4 illustrates the target service level metric Mt 401 with respect to the service level metric bound Mbounds 307 and the green belt 405. Mt 401 is the target value that falls within the green belt 405. The upper bound on the green belt 405 is Mhighbound 311 and the lower bound is Mlowbound 312. The green belt 405 is also bounded by Smin# 314 and Smax# 313. Thus, the green belt 405 is a representation of an SLA of the form (Smin#, Smax#, Mbounds).

An object of the dynamic resource allocation according to the present invention is to keep the operation state within the green belt 405. When the current operation state that is denoted by (M,N,R) is at 403 in the space 305, the primary operation is to reduce the currently allocated amount of resources N to the target amount Nt, so that the service level metric M at 403 would move to the target metric Mt at 404.

When the current operation state that is denoted by (M,N,R) is at 402 in the space 302, the current resource N may be increased to Nt when some free resources are available for allocation, or the inbound traffic R may be reduced to Rt so that metric M at 402 would move to Mt at 404. When the current state is within the green belt 405, no action is taken. The green belt 405 therefore defines the allowable system operation state region such that any state within the green belt 405 meets the service level agreement (SLA).

Figure 5 illustrates an overall system 500 according to the present invention including a main system 501, an inbound traffic controller 506, and a server resource manager 509.

The main system 501 includes a decision module and methodology 503 (e.g., algorithm), a module 502 (algorithm) for computing targets Nt(i) and Rt(i), and a repository for storing Service Level Agreements (SLA) 504.

The module 502 computes the target values Nt(i) and Rt(i) from the monitored data M(i) 204, N(i) 205 and R(i) 206 for every customer whenever its operation state (M(i),N(i),R(i)) falls outside of the green belt 405 associated with the customer.

Then the decision module 503, using the SLA information, (M(i),N(i),R(i)), Nt(i) and Rt(i), decides what action to take.

That is, the decision module 503 decides either to change the current resource amount from N(i) to Nt(i) 508, or bound the current inbound traffic rate R(i) by Rt(i) 505, and then take appropriate action.

System 501 has a communications means to instruct "server resource manager" 509 to change resource allocation 510. The system 501 has a communications means to instruct "inbound traffic controller" 506 to bound the incoming traffic 507 to a specific customer site (201 or 202).

Tables 2 through 5 give various means in computing or deriving target values Nt(i) and Rt(i) for every customer i.

For example, Table 2 describes formulas for computing these targets when the service level metric M is the resource utilization U.

Table 3 describes a formula for computing these targets when the service level metric M is the average response time T. Here, the average response time was derived from the "M/M/m" multi-server queuing model.

It is noted that since the computation is used for the "hill climbing" optimization and is repeated periodically, and the amount of resources allocated or de-allocated at each step is

assumed to be very small compared to the amount of resources currently allocated, the use of "M/M/m" model should be quite acceptable even though the arrival rate might be different from Poisson and the job processing time may not be exponentially distributed. A major advantage of "M/M/m" model is that it offers the closed form formula as shown in Table 3.

Table 4 describes formulas for computing these targets when the service level metric M is the response time percentile T%. Again, the "M/M/m" queuing model is assumed in computing the targets.

Table 5 shows that, instead of using a formula to compute the targets (Nt,Rt), one could use any numerical computation tool or quick simulation tool.

Figure 6 describes the decision method 600 employed by module (algorithm) 503 for server resource allocation in the system 501.

The decision method 600 looks for (e.g., attempts to obtain) potential revenue maximization opportunity when allocating free resources to various customers. It first seeks any opportunity to de-allocate resources, next allocates additional resources to customers whose service level metric is outside of the green belt 405 (Figure 4) and finally looks for when the customer's inbound traffic must be throttled (reduced) due to exhaustion of free resources or the maximum amount of resources has been already allocated.

Method 600 begins at step 601. In step 602, the target values (Nt(i),Rt(i)) are computed for every i. Further, the variable "ITC-informed(i)" = "no" is set for all "i". This variable keeps a record of whether or not throttling on inbound traffic has been applied or not prior to the current computation. This computation or examination is performed periodically to check whether or not any service level agreements have been violated, that is, checking whether or not any operation states falls outside of green belts. An examination is conducted in a time interval called a cycle-time. A cycle-time is a system operation configuration parameter. For example, a cycle

time value could be selected from a value between 1 second to 60 seconds. Whether to choose a smaller value or a larger value depends on how fast one can adjust resource allocation/de-allocation.

In step 603, it is determined whether or not the service cycle time has expired. If it has expired (e.g., a "YES" in step 603), the process loops back to step 602.

If "NO" in step 603, then in step 604 it is checked whether the operation state M(i) is within the green belt 405 (e.g., see Figure 4).

If so (e.g., a "YES"), then step 605 is executed in which the system waits for the cycle time to elapse and the process loops back to step 602.

If "NO" in step 604, then in step 606, it is checked whether any customer exists such that the target resource amount Nt(i) is less than the current amount N(i) (i.e., seeking an opportunity to de-allocate server resources from customers and placing them back into the pool of "free" resources).

If "YES" in step 606, one possibility that Nt(i) is less than N(i) is that because the inbound traffic has been throttled. This condition is tested at step 607. Step 606 identifies all those customers such that Nt(i) is less than N(i). Step 607 is applied to only those customers identified in step 606. Step 607 checks if there is any customer whose inbound traffic is currently throttled. If step 607 is "YES", step 609 is executed. Step 609 issues a command to ITC 506 to stop applying the throttling on the i-th customer's inboud traffic. and sets ITC-informed (i) = "no".

When Nt(i) is less than N(i) ("YES" in step 606) and the inbound traffic is not throttled ("NO" in step 607), that means that too many resources have been allocated to the given amount of inbound traffic for the i-th customer traffic, step 608 seeks to de-allocate resources away from the i-th customer.

In step 610, it is checked whether the resource(s) must be increased for any customer identified in step 606. There is no action required for those customers whose target value Nt(i) is equal to the observed value N(i). Step 610 identifies a customer whose server resource must be increased.

If so ("YES" in step 610) and if free resources are available ("YES" in step 611), then step 612 is executed to allocate additional resources (e.g., allocate up to Nt(i) - N(i) resources without exceeding Smax#(i)).

When additional resources must be allocated, and yet no free resource is available (e.g., a "NO" in step 611), then it is necessary to "re-claim" resources from those customers who have more than the guaranteed minimum (e.g., N(j) > Smin#(j)) (step 614).

When additional resource(s) must be allocated ("YES" in step 610), and no free resource is available ("NO" in step 611) and if the currently allocated resource N(i) is more than or equal to the guaranteed minimum Smin#(i) ("NO" in step 613), then the inbound traffic must be throttled (step 615). That is, the inbound traffic controller 506 is instructed to bound the traffic by Rt(i), and ITC-informed(i) is set to "YES".

As described above, with the unique and unobvious features of the present invention, a dynamic resource allocation is provided to a plurality of customers to meet with the (min,max) server resources and performance metric-based service level agreements.

When describing the embodiment of this invention, often a fixed size unit of allocable or de-allocable resources were assumed. However, one can easily generalize to the case where each allocable unit has a different amount.

Further, it is noted that the method of the invention may be stored on a storage medium as a series of program steps, and can be executed by a digital data processing apparatus.

While the invention has been described in terms of a preferred embodiment, the invention is not limited thereto and those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.